TRAKCESS 3.0
"Access to Tracks"

For the TRS-80 Model III

by
Roxton Baker

TRAKCESS was written with these two goals:

To bring under direct user control, with as much software assistance as is practical, every capability of the TRS-80's 179X Floppy Disk Controller (FDC).

To combine these capabilities into powerful "intelligent" functions whose only limitations are those of the machine in general, or of the FDC in particular.

THE ALTERNATE SOURCE

TRAKCESS 3.0

Table of Contents

BACKING UP YOUR MASTER TRAKCESS DISKETTE

Do NOT use your master TRAKCESS diskette! Always, always work from a backup disk. Follow the procedure below to make a working backup copy of the TRAKCESS program, and store your master disk in a safe place.

TRAKCESS is distributed on a "File Transfer Utility" diskette that will boot on either a Model I or Model III system. This utility (FTU) splits the disk into two parts - approximately one half is a Model I compatible format and the other half is Model III compatible format. There are two directories on the diskette, one for each Model. If you boot the disk on a Model I system, you will find that there are no programs stored on the Model I half of the disk, and the directory is empty.

It is intended that you boot the master TRAKCESS diskette on a Model III system - place the disk in drive zero, and press the orange "reset" button located on the keyboard to the right of the number pad. The screen will clear for a moment, and then you will see the "File Transfer Utility - by Kim Watt" message at the top of the screen. There will be brief instructions on using the utility, as well. In the middle of the screen, you should see the programs available on the diskette; namely, TRAKCESS and TRAKCESS/CMD. The utility will tell you that TRAKCESS is xx granules in length, and that TRAKCESS/CMD is xx granules in length. At the bottom of the screen, there is a prompt asking you for the number of the drive containing the destination diskette.

To fully utilize TRAKCESS, two drives are required. For this reason, we will assume that you have two (or more) drives. The File Transfer Utility will copy to a destination disk that is placed in drive zero, but it is much less of a hassle to copy to a diskette that has been placed in drive one (no swapping of diskettes will be required).

It is important that the files be transferred to a TRSDOS 1.3 diskette. TRSDOS 1.2 should NOT be used - if this is the only DOS that you have, we strongly recommend that you visit the nearest Radio Shack store and upgrade your 1.2 disk to a 1.3 disk. Once the files have been transferred to TRSDOS 1.3, you may then transfer them to your favorite operating system (NEWDOS/80, LDOS, DOSPLUS, MULTIDOS, or whatever). To do this, you will need to follow instructions contained in the manual for the particular DOS in use.

The File Transfer Utility provides the number of granules that each file occupies. Total them together, and locate a TRSDOS 1.3 diskette with at least that much FREE room (the number of FREE granules is displayed at the end of each DIR command). Place this diskette into drive one of your system. Answer the FTU destination drive prompt by pressing the "1" key. The FTU will prompt you to mount the destination disk and key ENTER. Press ENTER when the disk is properly mounted and the drive door closed. Make sure the destination diskette is not write-protected!

After you press the ENTER key, FTU will go and check the diskette and make sure there is enough room to copy the files. FTU will then copy them to the TRSDOS disk, will update the destination directory, and will

then prompt you to "Key <ENTER> to boot drive zero". At this point, take the master TRAKCESS disk out of drive zero, put it back in the diskette jacket, and lay it aside. Take the TRSDOS disk out of drive one, and place it into drive zero. Press the <ENTER> key to boot the TRSDOS disk.

When you see the TRSDOS READY prompt, type in DIR and press <ENTER>. Somewhere in the directory should be the two files TRAKCESS and TRAKCESS/CMD.

The TRAKCESS program should theoretically work under any disk operating system you own, except for CP/M. We have tested it under LDOS, NEWDOS/80 and TRSDOS. If you have any problems with any DOS, please let us know about them.

The master disk that you have received is not easily accessible if you have only 80 track disk drives. If this is the case, we will need to prepare you a special master disk, and it will have to be a Model I double density formatted disk. Every popular operating system will be able to read the Model I format, so this should present no problem. Return the 40 track disk that we sent you, and specify which DOS you would like us to use when formatting the special 80 track master disk. Please include a postpaid return mailer (sturdy enough to prevent diskette damage) or $3.00 for packaging and return postage.

TRAKCESS 3.0
"Access to Tracks"

For the TRS-80 Model III

ROXTON BAKER
P.O. Box 8272, APO San Francisco, 96555

Congratulations! You have just purchased THE MOST POWERFUL TRS-80 disk access utility yet written. Should you discover any problems with the program, or have suggestions for improvement, please either contact The Alternate Source or write to me. User feedback is one way that good programs get better. Your input will be appreciated.

INTRODUCTION

Several fine disk access programs are currently available, such as Superzap, Z80ZAP, and Super Utility. For zapping standard DOS disks they are excellent, featuring quick access to a file's sectors and convenient "paging" through them. TRAKCESS is NOT intended to replace these utilities. It is unable to interpret a disk's directory, and since it makes no distinction between "standard" sectors and any others, it cannot offer the pre-formatted display and easy file I/O of, say, Superzap. TRAKCESS was written with different goals:

* To bring under direct user control, with as much software assistance as is practical, every capability of the TRS-80's 179X Floppy Disk Controller (FDC).

* To combine these capabilities into powerful "intelligent" functions whose only limitations are those of the machine in general, or of the FDC in particular.

The inspiration for this work was Bill Barden's excellent and unique Disk Interfacing Guide for the TRS-80, now available from 80-US Publishing, 3838 South Warner St., Tacoma, WA 98409 (or from TAS). Another excellent reference is Machine Language Disk I/O & Other Mysteries by Michael J. Wagner, published by IJG, Inc. This book is also available from The Alternate Source. You will need one of these books, or similar, if you are not already an expert on the FDC. This documentation will briefly discuss the elements of track formatting and sector identification, but AT A MINIMUM the non-expert will require a copy of the Western Digital data sheet on the 179X FDC (now being included in this guide).

TRAKCESS is powerful precisely because it works on the most elementary levels. It has been padded around with lots of interactive software and

utility functions so that it remains fairly efficient to use, but it is not the tool of choice for simple zapping. TRAKCESS assumes very little about the disks it is used on. As a result there are often many questions for the operator to answer. To make full use of TRAKCESS you must be prepared to study the literature and experiment with the possibilities. I recommend taking a few track dumps of disks you have on hand and comparing what you find there to what the books say. Then try making up your own tracks and formatting a blank disk with them. TRAKCESS puts all of this at your fingertips. Beyond this, you might exercise the disk identification features such as (S)can Track Sectors and (L)ocate Disk Sectors. These commands embody some routines that are simple conceptually but that were quite difficult to write. They are far more potent than they seem and form the heart of the highest level commands in TRAKCESS, such as (C)opy Track and (D)uplicate Disk. These latter are capable of analyzing many widely different disk formats and duplicating them, and are far removed from the primitive FDC functions. However, in the event that they require your assistance, you will still have to know what you're doing.

USING TRAKCESS

TRAKCESS requires a 48K machine. The (C) and (D) commands require two drives. The program has been tested under LDOS, NEWDOS/80, and TRSDOS, but should work under any Model III DOS. TRAKCESS consists of:

TRAKCESS	-	a BASIC program, and
TRAKCESS/CMD	-	a machine-language file.

To get started from DOS, just type 'TRAKCESS'. Then follow the prompts. When going into BASIC it is a good idea to specify 0 files, since TRAKCESS may need the extra memory.

IMPORTANT NOTE: Typing TRAKCESS from DOS READY is not optional! Doing so loads the machine language portion of the TRAKCESS program into memory so that the BASIC portion of TRAKCESS can access it. Many users have misinterpreted this step to be "just a copyright notice" or "just to tell me what to do", and then figure they can skip it. This is not so, and the BASIC program "TRAKCESS" will not run properly unless the machine language portion is properly installed in memory.

At the query as to whether a printer is attached and ready, your Y or N answer will determine whether or not line-printer output options will be offered to you later. All line-printer output is sent to the screen as well. After that, the menu will come up.

IMPORTANT NOTE: at almost any time when the program is awaiting input or is executing a lengthy command, pressing 'M' will return you to this menu.

Most prompts identify their default values with an '*'. Thus the prompt (*Y/N) indicates that pressing <ENTER> is the same as pressing 'Y'. You will also notice that some of the default values in the program are "adaptive"; that is, they change to reflect the values you used last.

TRAKCESS expects memory addresses to be entered as four hex digits, without a trailing 'H'. When entering two-digit (byte) values, it is generally not necessary to include a leading zero - TRAKCESS will supply it for you.

The following is a description of each of the menu commands available. Please read this at least briefly before running the program. TRAKCESS, like any other disk utility, can easily destroy valuable data on a disk if you aren't reasonably careful. ALWAYS write-protect your master disks, and work on back-up copies!

Some terminology used in these descriptions will not be familiar to you. Definitions and explanations will be found in the DISK STRUCTURE AND FORMATTING section.

1. Select Drive

You must tell TRAKCESS which of the four drives is to be the currently

active one. You'll be asked its head position. Normally, you won't know, so just press <Enter>. If the drive was previously selected TRAKCESS will recall its last position, otherwise the head will be moved to track 0. If you enter a number here, TRAKCESS will assume that is the actual head position of that drive. NOTE: no other disk-related commands will work until a drive has been selected!

TRAKCESS contains all of the disk I/O routines that it requires, so once it is running you do not need a DOS disk in drive 0. However, if TRAKCESS stops on a program error, put a DOS disk in drive 0 before continuing in case BASIC looks for one. Otherwise you may hang the system.

2. I and O

These step the head of the selected drive In and Out one track. "In" means closer to the center of the disk. These two keys repeat. TRAKCESS assumes forty tracks. This is defined by DM (presently set at 39) near the beginning of the program. If you wish to work with a different number of tracks, change DM to that number minus 1.

3. (G)o to Head Position

Allows you to quickly position the head of the active drive at any track. Enter the desired track location in decimal.

4. (R)ead and (W)rite Sectors

These commands allow you to read into a specified block of memory, or write from it, any sector on the current track. You must specify the starting point in memory, the sector's ID bytes (TN and SN), and its DAM (for a write). The length of the sector is, of course, already noted on the track, having been put there at format time. If a sector was previously specified TRAKCESS will allow you to reuse those specifications without reentering them. This facilitates modifying sectors and transferring them between disks.

One further feature of sector (R)eading is useful in the recovery of lost data. TRAKCESS will report an unsuccessful read and will give you the option of infinite retry. Unlike other utilities, TRAKCESS does not step the head all over the disk between attempts. That seems so hard on the drive. I have had TRAKCESS recover a "lost" sector on the 450th try! All sector (W)rites are verified by rereading, so data transfers under TRAKCESS are reliable.

5. (T)ake and (P)ut Tracks

These two commands involve the transfer of a whole track (from 3120 bytes to 6250 bytes, depending on density) of information. A track read followed by a scan with TRAKCESS' editor will show you literally

everything that is on the track. Do not be concerned if the number of bytes read in varies from time to time. This is normal for a track read. Also, you should not expect to see the track's sectors all neatly lined up togethev. Track reads show you all of the filler material and sector ID blocks, in addition to the sectors themselves. In fact, even if a track is "empty", there will be random magnetic patterns which will be read into memory by the (T)ake command.

The track write is usually used only for formatting a blank disk with empty sectors. You can create any format track you want with the (B)uild command, then write it to disk with (P)ut. If none of the sectors on a track contain F7 bytes, it is feasible (though not always successful) to (T)ake a track from one disk and immediately (P)ut it onto another. Synchronization is a real problem here, but it can work. In fact, sectors with F7 bytes can then be transferred separately using (R) and (W). TRAKCESS will do these things, but considers them poor form. You would only attempt to copy a track in this manner if you were desperate, and afterwards you wouldn't tell anyone.

The (P)ut command also allows creation of a track which contains both single and double density sectors. Information on how this is done is discussed later in the APPLICATIONS section.

6. (S)can Track Sectors or (L)ocate Disk Sectors

TRAKCESS will search the current track (or the whole disk) for all useable sectors. It will determine all the important information for each (TN,SN,SL,DAM) and will present it to the screen or printer. Do not be impatient; up to 30 seconds per track may be required for the scan. Try these commands on a standard TRSDOS disk, any "protected" disk, a CP/M disk, or even an Intecolor disk if you have one, and compare the results.

If a track has any "false" sectors (without a data record following, or with a bad ID pack) TRAKCESS will note the fact. If you are attempting to recover a damaged disk, you may be able to find some sector data using a track read, and transfer it to a track you have built using the (B)uild format track command discussed below. Or the false sectors may have been purposely put on the disk as some sort of "protection". If you are attempting to duplicate such a "protected" disk, use the (B) function and the editing technique outlined there.

Note that the (L)ocate command first tells you which tracks even have sectors, and then gives you the option of a full report. This command is designed to determine the density of the track, as well as sector content. In the summary report each track which was found to contain sectors is listed. Each track number is followed by 'S' for single density sectors, 'D' for double density sectors, or 'B' for both single and double density sectors on the same track. Please note that only (L)ocate works this way. (S)can will detect sectors recorded in the current density (See paragraph 13) ONLY, even if (S)can is invoked from (L)ocate.

7. (C)opy Track

You are prompted for a drive and track to copy to. You cannot copy to the active drive; another must be specified. TRAKCESS will search the current track and will identify all the sectors on it. Then it will build a matching format track in memory and write it to the target disk. Finally, it will transfer (with verify) all of the sectors. You are given the option of using the existing DAM's or specifying a new one. If any sectors on the source track are damaged or of indeterminate length, you will be prompted for the length you wish to make them. To decide on this, do a track read and look at that sector in memory. Try choosing the next larger multiple of 128 bytes than the sector's apparent length, unless you know what it should be.

The (C)opy function will also allow you to specify that the target track on the destination disk NOT be formatted. This option is useful when you have already created a format track and have (P)ut it onto the destination disk, and wish only to copy sector information over from the source disk. If you do select "No Format", be aware that sectors with bad lengths and/or bad CRC checks will not be transferred. You will usually want to handle these manually anyway.

8. (D)uplicate Disk

This command does nothing more than set up both drives and repeat the (C) command for each track. Most disks, "protected" or not, can be duplicated automatically in this fashion, but it usually takes at least 15 minutes. It's a good idea to check out the destination disk first by formatting it using TRSDOS, and then bulk-erasing it. Note that unlike Superzap et. al., TRAKCESS will copy to a blank disk as well as to a formatted one. If any sectors won't copy, note them and proceed. You may later be able to use (R) and (W) to transfer them, or that entire track may be retried with (C).

(D)upe gives you the option to copy the sectors with or without first formatting the destination disk. Normally a format will wanted, and this is the default condition. In some cases, e.g. dual density diskettes, it may be desired to do the copy without format. Note that if this option is selected the destination disk MUST have been previously formatted with all the necessary sectors for the copy.

The (D)uplicate command gives you the option of specifying that only certain tracks be copied. These tracks will be transferred into their corresponding locations on the destination disk. During the selection of tracks to be copied you may press 'C' to indicate that all remaining tracks are to be copied, or 'S' to indicate that all remaining tracks should be skipped. Note that any tracks skipped on the destination disk will not be erased - they are left as-is.

9. (E)dit or Fill Memory

This command allows you to edit memory with the scrolling, Electric Pencil-like editor utility that is built into TRAKCESS. This editor is called PENRAM. Or you can fill memory between specified addresses with a specified byte. The editor requires a four-digit hex starting address. Upon entry of this address, a hex display of the 256 bytes of memory beginning at that address will appear on the screen. You may change this to an ASCII display by pressing the <CLEAR> key. Pressing the <CLEAR> key again will return you to a hex display. The blinking edit cursor will be in the upper left corner. You move it using the four arrow keys - any scrolling required is automatic.

On the right is continuously displayed the current address of the blinking cursor, and below that a "reference address" and the number of bytes (decimal) that the cursor is offset from it. Pressing shift/BREAK will reset the reference address to the current cursor address. To understand the use of the reference address feature, imagine that you wish to move the cursor 287 (decimal) bytes beyond where it is. You would press shift/BREAK to set the reference address to the current position of the edit cursor, and then you would move the cursor downwards while watching the displacement value to see when you've reached 287.

The arrow keys move the cursor; shift/UP-ARROW and shift/DOWN-ARROW scroll rapidly. Under either hex or ASCII mode, whatever you type in will be put into memory at the cursor location (an "overwrite" mode, rather than an "insert" mode). When replacing data, two hex characters (0-F) are required; in the ASCII mode any printable character may be entered (except arrows). The characters entered will replace the current byte at that location. You may press <BREAK> after entering only one hex character, in which case the original byte is restored. All data changes are seen instantly on the screen. Most keys repeat so that you may easily fill memory with a value, and high-speed scrolling is provided with the shift/UP-ARROW and shift/DOWN-ARROW keys. A little practice will clarify these instructions. The editor is much easier to use than to explain. It is one of the few utilities in TRAKCESS that will not return you to the menu via 'M'. You must press shift/LEFT-ARROW.

10. (B)uild Format Track

This powerful command allows you to specify interactively a track to be created. There are no restrictions other than overall length (6250 bytes) on the track contents. You may specify up to 50 different sectors, of differing lengths and types, with any names. You are allowed to change the specifications on any sector; when finished, the format track may be created (at E000H+) and (using P) written to any track of the disk. This command is very useful in creating disks that can more efficiently store and retrieve the types of data you wish to work with. For example, you might create a track with only one sector of 1024 bytes. This entire block could be read into RAM quickly using a short sector read routine as explained in Barden's book. You might also

use (B) to hand-tailor a track to match one on a "protected" disk.

To specify a sector you must supply a TN, SN, and SL. A typical entry would be:

09,2D,03

This specifies a sector with TN=09, SN=2D, and SL=03. The SL value indicates that the sector will be 1024 bytes long. The next sector might be specified as:

C1,77,02

resulting in a sector with TN=C1, SN=77, and SL=02. As before, its length is calculated by multiplying the SL value (which is in hex) plus one by 128 bytes. This sector would be 384 bytes long. These two sectors may or may not be the only ones on this track.

On entry to the (B)uild command, TRAKCESS checks to see if a valid set of sector specs already exists. This will be the case if you have previously done a track (C)opy or (S)can. You may use these specifications if you wish. In fact, this is a good way to review the characteristics of a track you previously (S)canned - go into (B)uild and list the specs. If the track contained false ID packs, however, the (B)uild command will not allow you to reuse the specifications.

Also, you will note that when specifying sector names, the (B)uild command will allow you to just press <ENTER> to add one to the previous sector's SN, keeping TN and SL the same. This lets you build a track of similar, consecutively-numbered sectors very quickly. (B)uild will also let you change all TN's or all SN's to a specified value. This makes it easy to create successive tracks in which only one of these values changes (as on a standard TRSDOS disk).

An important point about (B)uild is that it ALWAYS builds the track according to the rules for the current density. Simply by changing density, the same set of sector specifications could generate two completely different tracks. (B)uild normally starts construction of track data beginning at E000H. To facilitate (B)uilding of tracks containing both single and double density sectors, it is possible to change this (S)tart address, thus preserving everything between E000H and the (B)uild starting point.

Once you have created the format track at E000H+, you may wish to edit it directly in memory before writing it to disk with the (P) command. Do this by returning to the menu and using the (E)dit function to work at E000H. One reason you might do this is to zero out the DAM for a sector that you have created, thus generating a "false sector ID"; that is, one with no data after it. Or you might wish to take out the F7 bytes that are put into the format track after every ID pack and sector record. This F7 is what causes the FDC to write two CRC bytes to the disk when it is encountered. Taking it out will result in a false ID pack (if you take out the ID CRC) or an always-bad sector (if you take out the sector CRC). Such unproductive shenanigans will only be

necessary if you are breaking a "protected" disk.

Once a track has been (B)uilt in memory it may be desirable to (P)ut it to more than one track on disk. In such cases it will be necessary to change the track numbers first. This may be done by (E)diting the buffer, but that is a cumbersome and tedious method. Using the (I)ncrement option will automatically search the buffer for FE bytes and add 1 to the value following them.

The (B) command is probably the most fun thing TRAKCESS offers, next to stepping the head back and forth.

11. (F)igure CRC's

This command will allow you to calculate the two CRC bytes for any block of code in memory, or for any bytes you type in. This is useful when examining damaged or tricky sectors or sector ID's. The CRC algorithm used by the FDC is the IBM SDLC standard. It it sufficient to regard the CRC as a two-byte checksum of the data it's applied to - a checksum that is always initialized to FFFF (single density) or CDB4 (double density). The FDC starts calculating a CRC whenever an Address Mark of any kind (F8-FE) is encountered during a track write, and whenever a sector is being written or read. A subsequent F7 (on a track write), or the end of the sector, cause the FDC to spit out the two current CRC bytes. On any write, these go to the disk. On a sector read, they are reported back to the CPU, along with an indication of their correctness (as compared to the CRC bytes on the disk). Consult the references for more details on these sequences.

12. (H)ex dump to the printer

This actually gives you a choice of hex or ASCII dump, starting at any address. Like all the other printer options in TRAKCESS it will only be offered if you have said that a printer is ready. It will break to the menu if 'M' is pressed.

13. (A)lternate current density

TRAKCESS always begins in double density, since that is the "native" state on the Model III. However the 179X FDC is capable of both single and double density operation. The current state is always shown at the top of the menu screen following the current drive and track location. This command simply toggles between single and double. One must be very careful, since the outcome of many commands will vary, depending on the current density. See APPLICATIONS for more information.

TRAKCESS has a lot of error-checking built into it on operator inputs, so you should never accidentally crash it. Also, most requests for input have default values specified - when in doubt, push <Enter>! Remember that entering 'M' will usually take you back to the menu.

One common question about TRAKCESS is why it will not "page" through sectors a la Superzap. As mentioned earlier, this is because TRAKCESS does not assume anything about the next sector on a track. There may not even BE a next sector! The lack of such a feature is one of the performance/convenience tradeoffs that must be made in a program like TRAKCESS.

When driving the printer, TRAKCESS uses only BASIC's LPRINT command, so it should be compatible with any printer you are using. If you have a software printer driver or other program in high memory, remember that TRAKCESS uses all RAM up to F8FFH. Programs above that are safe from alteration unless you specifically tell TRAKCESS to load something in there. This space has primarily been left so that you may drivers for peripherals in memory at the same time as TRAKCESS. Load any such program before running TRAKCESS. Unfortunately, future versions of TRAKCESS will not be able to leave much, if any, of this additional space free. There will, however, be compensating advantages. This is why it is important for you to remain informed of upgrades to TRAKCESS!

If you experience an Out of Memory error during the execution of TRAKCESS, try decreasing the amount of string space CLEARed near the beginning of the program. Also be sure that you haven't accidentally specified files, as TRAKCESS does not need any! And note that TRAKCESS sets its own memory size when first called, by placing a value into 4411H. This will work with any recent DOS. If your DOS does not respect this value, you will find that BASIC eventually overwrites the TRAKCESS program, causing syntax errors, undefined lines, etc.

Another possible area of trouble is in your hardware. Make sure that you have all the latest Radio Shack fixes to your Expansion Box (if it's an old one), and that the card-edge connectors are not dirty. They may be cleaned with a pencil eraser.

DISK STRUCTURE AND FORMATTING

The characteristics and capabilities of the 179X Floppy Disk Controller completely determine the track structure of any TRS-80 disk. The data sheets mentioned earlier fully describe these capabilities, and Barden's book explains the overall disk layout and the integration of the disk drive and controller with the TRS-80 CPU. What follows here will cover the middle ground. There are some things in the data sheets that do not apply to the TRS-80, and there are many things that can be done that are not discussed in Barden's book (since his book addresses only standard TRSDOS formatting).

The actual number of tracks on your disks is determined by the capabilities of your disk drive, not by the FDC, and is irrelevant here. As detailed above, you can make TRAKCESS handle any number of tracks. What is important is the contents of each track.

For a track to have any easily-accessible, reliable information on it, it must be set up to hold that information in one or more sectors. A "sector" is an area on the track that consists of an ID block, followed by some filler bytes, followed by a data block. There are also filler bytes before and after the sector area. In general, these filler bytes are strings of FF's and 00's. It is necessary for them to be present, in many cases even to an exact byte count. Following these filler bytes are "track number", "side number", "sector number", and "sector length". I will refer to these from now on as TN, SD, SN, and SL, respectively. Each of these bytes may have any value from 00 to FF, except for the restricted values F7-FE. It is important to note that REGARDLESS of what disk track a sector is actually, physically on, it may have any of the allowable values for TN, SN, SD, and SL. Do not confuse TN with the number of the track that the sector is on! Do not confuse SN with the position on that track of the sector! The first sector on track 3 could very well have TN=C2, SN=1A. There is not necessarily any correlation. It so happens that standard TRS-80 disks DO make the TN equal to the actual track number, but tlap is merely a convention.

The value of SD is quite unimportant to TRAKCESS. It is always set to 00 and is never compared with the side bit in the FDC. TRAKCESS does not presently allow you to specify any value other than 00 for SD. Forget about it.

That leaves TN, SN, and SL as the important sector specifications that appear in the ID block. The two bytes TN and SN essentially comprise the "name" of the sector. No two sectors on a track may have the same name (the same TN, SN pair); if they do, the data in the second of these sectors will not be accessible. Aside from their function as a name, TN and SN mean nothing.

The length of the sector's data block, hereafter referred to as the sector's length, is specified by SL. The length of an IBM sector is either 128, 256, 512, or 1024 (dec.) bytes, corresponding to an SL value of 00, 01, 02, or 03. These are the only four allowable values for the SL of an IBM sector. Unlike the 1771 FDC used in the Model I, the 179X family can read and write ONLY IBM sectors. Standard Model III TRSDOS

disks have, as you will find if you do a (S)can on one, eighteen IBM sectors on each track. Each sector is 256 bytes (SL=01).

The ID block of every sector also contains, as its first byte, an FE. This immediately precedes the TN. This FE is called an ID Address Mark, or IDAM. It and the four bytes that follow it (TN, SD, SN, SL) are referred to as the ID pack. These are followed by two CRC bytes as described above under the (F) command. These CRC bytes serve as a checksum on the ID pack bytes. The IDAM alerts the disk controller that a sector is coming up. The FDC reads in and verifies (by checking the CRC bytes) the rest of the ID pack, thus determining the sector's name. If this is the sector it has been told to read or write, the FDC then calculates the sector length from the SL value and from its knowledge of the sector type. This done, the FDC awaits the appearance of the Data Address Mark, or DAM. The DAM byte may be either F8 or FB. The Model III TRSDOS standard value is F8, except for DOS directory tracks which get an FB. Every sector data block, or "record", must be preceded by a DAM to be recognized. The record must be followed by two CRC bytes, which here serve as a checksum on the DAM and the data in the data block. On a sector read, knowing the length of the record, the FDC now reads it in, all the while computing the CRC bytes which it finally checks against those at the end of the record. The FDC will return various error messages if it encounters bad ID pack CRC bytes, a missing DAM, or bad sector record CRC bytes. On a sector write the DAM must be specified. Again knowing the length, the FDC writes the specified DAM out over the existing one, and then writes out the correct number of data bytes to fill the record. All the while it is computing the record's CRC bytes, which it finally writes out over the existing two CRC bytes at the end of the record. Here too the FDC will return various error messages if required. These error messages are returned as codes in the FDC status register, and are detailed in the references. TRAKCESS will interpret the most common errors for you; if it can't make sense of the error encountered, it reports the code that was returned.

The above description pretty much covers the makeup of each individual sector on a track, the characteristics you may find, and the reasons for them. The (B) command of TRAKCESS may be used to create a few tracks, and the (R) and (S) commands will read and write the sectors on them. Remember that the (B) command creates tracks with "empty" sector records, filled only with E5's. You must write this "format" track off to disk, and then fill each sector with sector writes. Experimentation with these commands should answer most questions you may have concerning track structure. The references, especially the FDC data sheet, do contain almost anything you could wish to know; finding it in there is the only problem.

Like most powerful tools, TRAKCESS requires time to be used to the utmost. The purpose of this section is to get you started with some examples, and to provide some helpful tips and guidelines.

Example 1 - Repair a Disk Sector

Perhaps the first thing one might wish to do with TRAKCESS is to repair a damaged diskette. For illustration purposes let us assume that we have a Model III TRSDOS disk whose backup counter has expired, preventing us from making further backups of the disk. Repair of this problem requires changing only a single byte in the disk boot sector.

Step 1 - Select the drive

Insert the disk to be repaired into drive 0. From the TRAKCESS menu push '0' (the zero key, not the letter). TRAKCESS will ask for the head position (unless you have previously accessed this drive). Simply push ENTER. Drive 0 is now selected. Wasn't that easy?

Step 2 - Position the head

The banner above the menu should read as follows:

```
* HEAD 0 IS AT TRACK 0  DOUBLE *
```

Yours doesn't look that way? Don't panic! If it doesn't say HEAD 0 then go back to step 1. To (G)o to track 0 type 'G' and answer with 0 as the track to go to. Last, if you are in single density just type 'A' to (A)lternate to double. Now check the banner again. If everything looks alright proceed to step 3. Otherwise try this step again.

Step 3 - Read the sector

Now we're ready for the big stuff - (R)eading a sector! Take a deep breath and push 'R'. Next you will be asked for the Track Number. Since we are on track 0, and it is the default, you may just push ENTER. If you're not sure, then type in 00 and push ENTER. You must enter the sector number, since there is no default. Type 01, followed by ENTER. The next screen will ask you where to read the sector into memory. Push ENTER to use the default value of E000. Now in a few seconds we will get a report indicating that the sector has been successfully read into E000 for a length of 256 bytes. If yours was not successful try to figure out from the error message what the problem is and try again.

Step 4 - Fix the data

Once the sector has been read we are give the opportunity to look at the data by pushing 'E'. Do this now. Wow! Did you ever see so much stuff on one screen?! This is the sector in hexadecimal form. If you don't know what hexadecimal is you are probably in over your head -- take a break until you have read up on it.

The RAM editor is a little program called PENRAM, which works a little bit like the Electric Pencil word processor. You use the arrow keys to move the cursor (that little vertical line) around on the screen. Instructions for PENRAM are included elsewhere. Move the cursor to E022 (watch the address in the upper right hand corner). The byte there will have a value of 00. Type in FF. You have just changed memory. We are nearly finished!

Step 5 - Rewrite the sector

Exit PENRAM by pushing the SHIFT and left arrow keys together. From the menu to 'W' to request a sector (W)rite. Now you will be advised that a sector specification exists for track 0, sector 1. Answer 'Y' (or push ENTER) to use the same one. Likewise push ENTER to do the write from memory address E000. In a few seconds a message will be displayed indicating a successful write for a length of 256 bytes. The job is done!

This example was somewhat trivial, just to get you started. Now let's move on to the tough stuff.

Example 2 · Duplicate a dual density disk

It is quite common since the release of the TRS-80 Model III to buy software on diskettes which can be used interchangably on either the Model I or the Model III. This is accomplished by formatting some tracks of the diskette in both single and double density, so that either computer may read them in the "native density." Such diskettes are impossible to back up using conventional backup utilities. It is possible, if somewhat laborious, to back such diskettes up with TRAKCESS.

Step 1 · Select the drive

Place the diskette that you want to back up in drive 0. Make sure it is write-protected!! Put a blank disk in drive 1. From the menu type '0' and select the drive in the normal way.

Step 2 · Locate the sectors

Type 'L' to locate the sectors on the disk in drive 0. This takes some time. Let's suppose that when it finally finishes and you print the report, it looks like this:

```
TRACKS WITH POSSIBLE SECTORS:

   0B    1S    2S    3S    4S    5S    6S    7S    8S    9S
  10S   11S   12S   13S   14S   15S   16S   17B   18D   19D
  20D   21D   22D   23D   24D   25D   26D
```

What does this mean? Simply that tracks 0 and 17 contain (B)oth single and double density sectors. Tracks 1 through 16 are (S)ingle density only, while 18 through 26 are (D)ouble density only. Since these are the only tracks with sectors, they are the only ones we need to worry about.

Step 3 · Map the dual density tracks

Position the head over track 0. If the banner above the menu does not say SINGLE then push 'A' until it does. Next push 'S' to begin a sector scan. When complete we will get a report something like this:

```
              -- TRACK 0 --
SECTOR   TN   SN   SL   IBM   LENGTH   DAM
   1     00   00   01    I      256     FB
   2     00   05   00    I      256     FB
   3     00   01   00    I      256     FB
   4     00   06   00    I      256     FB
   5     00   02   00    I      256     FB
   6     00   07   00    I      256     FB
   7     00   03   00    I      256     FB
   8     00   08   00    I      256     FB
   9     00   04   00    I      256     FB
```

```
 10    00  09  00   I     256    FB
```

Now push 'A' until the banner says DOUBLE. Then push 'S' to scan the double density portion of the track. That report might look as follows:

```
                -- TRACK 0 --
SECTOR   TN   SN   SL   IBM   LENGTH   DAM
   1     00   01   01    I      256     F8
   2     00   03   01    I      256     F8
   3     00   02   01    I      256     F8
```

We now have a complete picture of track 0. The other dual density track is 17. Repeat step 3 for track 17 (don't forget to position the head).

Step 4 - Build a dual density track

When building a dual density track it is necessary to watch which density you are in very carefully. Being in the wrong density at the wrong time can cause all kinds of trouble. To start building track 0 be sure the density is SINGLE. Then type 'B' to request the (B)uild command.

(B)uild will ask for the number of sectors to be created. Since we are doing the SINGLE density portion, enter 10. Then enter the sector specifications as indicated, using the report from (S)can as a guide. When finished type 'E' to edit. You will note that the 10 sectors finish up somewhere after EBOOH. At location EBFFH enter the value F4H. This is VERY important! When the track is (P)ut the value of F4H will tell TRAKCESS to change density at that spot. When this has been done return to the menu.

Step 5 - Continue to (B)uild

Next exit (B)uild and use the (A)lternate command to go into DOUBLE density. Go back into (B)uild. This time enter the specifications for the 3 double density sectors. Before you build the track use the 'S' subcommand to change the start address to EC00. This will preserve the single density data already (B)uilt. Then build the single density track.

Step 6 - (P)ut the track

It is of utmost importance to switch to SINGLE density now. That's because the first 10 sectors in RAM are in single density format. If we (P)ut the track in double density those 10 will be written out in double, then the switch, then the last 3 in single density. The result will be a track with NO readable sectors! Remember, when (P)utting a dual density track you must start out in the proper density for the first sectors.
Mapping the double density portion of the track first is possible, but

may require many tries to (P)ut successfully. Single density first will nearly always work the first time.

Select drive 1 and (G)o to track 0. Type 'P' to select (P)ut. Use the default memory address of E000. The byte count that is written will vary, and has little meaning.

Step 7 - Check the new track

While still in SINGLE density do a (S)can to verify that all the necessary single density sectors are there. Then go to DOUBLE density and repeat the (S)can. If any are missing, go back to step 6. (P)utting a dual density track is a tricky and timing-critical thing to do, and it may not always work the first time. Keep repeating steps 6 and 7 until ALL sectors can be verified.

Repeat steps 4 through 7 for track 17.

Step 8 - Duplicate the disk

Select drive 0 and DOUBLE density. Request the (D)upe command, using drive 1 as the destination. For the duplication select only tracks 18 through 26, since only those tracks are ALL double density. When this is complete go to SINGLE density and (D)upe tracks 1 through 16.

Step 9 - Copy the dual density tracks

Select drive 0 and SINGLE density. Use the (C)opy command WITHOUT FORMAT to copy the 10 single density sectors on track 0. Then (A)lternate to DOUBLE density and copy the 3 double density sectors, again WITHOUT FORMAT. Repeat this process for track 17.

This concludes the backup process. As you can see it is long and involved. However, without TRAKCESS it would not be possible at all.

Regarding Previous Versions of TRAKCESS

All prior versions of TRAKCESS were designed to operate in single density only on the TRS-80 Model I. The 1771 FDC in that machine is capable of writing both IBM and non-IBM format sectors. Non-IBM sectors can range in length from 16 bytes to over 4,000 bytes. The 179X FDC chip family is unable to do anything with non-IBM format sectors. Therefore it is possible that some Model I single density diskettes will not be able to be backed up, or even read, using TRAKCESS 3.0. This is due entirely to hardware and cannot be overcome by software.

On (B)uilding Tracks

Using the (B)uild command it is possible to put almost anything on a track. The FDC doesn't care what you do -- it simply drops the bytes that are sent to it on the disk. At first glance it might appear that the byte counts and values between sectors are arbitrary. Nothing could be further from the truth. Successful operation of the FDC depends on its ability to recognize certain bit patterns. The value and number of the "filler" bytes between sectors has been carefully chosen to synchronize the FDC with the data on disk.

When (B)uilding your own formats pay very close attention to the minimum byte counts given in the 179X data sheet. If you try to cheat in order to get more data on a track you might find that you are suddenly getting intermittent errors. Don't blame your hardware or TRAKCESS -- that's the price you pay if you break the rules. This is especially true in double density, where the gaps are twice the size of those in single density. These larger gaps account for why double density gives you only 1.8 more sectors than single density.

The legendary E5 pattern in the sectors is also not arbitrary. In single density the E5 presents the worst-case bit pattern when trying to read a disk. Therefore E5 is used during formatting to weed out marginal and defective diskettes. Double density, which uses a different recording scheme, is not stressed by E5. The worst-case pattern for double density is a two-byte pattern of 6DB6. TRAKCESS, which was not designed as a diagnostic or test tool, always uses E5.

On Dual Density Tracks

The ability to format a track in both single and double density is a TRAKCESS exclusive. This is due to the characteristics of the 179X FDC and the way it is used within the Model III.

As previously noted, a value of F4 within a track format will cause density to be switched at that point. All bytes before the F4 byte will be recorded in one density, while all those after the F4 will be recorded in the other. The F4, and one or two bytes on either side of

it, will not be recorded.

When (B)uilding a dual-density track the placement of the F4 byte is very critical. TRAKCESS counts the number of 256 byte groups that precede the F4. When those bytes have been written in the original density TRAKCESS changes density and outputs the rest of the track. There are two important points here! First, only the first F4 has any effect. Any others will be written to disk like any other data byte. This means that density can be switched ONLY ONCE on a track. Secondly, since the first part of the track is written as multiples of 256 the F4 byte must be carefully placed. For example, suppose a track contains double density data from E000 to E425, followed by single density data. If F4 were placed at E430 then only 4 groups of 256 would be written (E4-E0). Placing F4 at E500 insures that 5 groups (E000-E4FF) will be written before the switch.

In general it is best to place the F4 byte at an address ending in FF. Be sure to leave 20-30 bytes of filler on both sides of the F4. That's because the density switch leaves a region of "noise" on the disk, which could hinder reading any ID packs written too close to it.

Switching density in mid-track is not something that the 179X FDC was specifically designed for. It is tricky and time-critical, and it may not work the first time you try it. Best results will be obtained if the single density portion of the track is written first. The double density portion may also be first, but many attempts may be necessary to write it successfully. Always use (S)can to insure that all the sectors in each density are properly recorded after (P)utting a dual-density track. If it consistently fails make sure that the INITIAL density is the same as the current density of TRAKCESS. If you start out in the wrong density neither one of them will be readable.

On (G)etting Tracks

The (G)et command allows an entire disk track to be read into memory. This is very different from a sector read, however. Only the ID packs and sector data should be considered reliable. During the reading of a track the "filler" bytes are subject to bit shifting due to the FDC being out of synchronization with the drive. For example, the 4E filler bytes of a double density track may be read as 9C.

Due to the nature of the track read it is also possible that one or more bytes may be missed by TRAKCESS, since the FDC my transmit data up to four times faster with this command. The designers of the FDC included the track read as a tool for engineers and designers, but did not expect it to be used as a standard thing. Neither should you!

On Model I Diskettes

The TRS-80 Model I uses a 1771 FDC, which operates only in single density. In addition to the non-IBM capability previously discussed, this chip also has the ability to write four different Data Address

Marks (DAMs). Unfortunately the 179X family used by the Model III is able to recognize only two of those. DAMs F8 and FB will be recognized and reported as such. FA will be reported as FB, and F9 will be reported as F8. It is not possible to write a DAM of F9 or FA with TRAKCESS. That means that you should not try to write a Model I disk directory sector from TRAKCESS, since the Model I DOS will no longer recognize it as having an FA DAM.